

Ejercicios Docker

Ejercicio 1

- Dockeriza la aplicación dentro de `lemoncode-challenge`, la cual está compuesta de 3 partes:
 - Un front-end en React.js (Node.js).
 - Un backend en .NET core.
 - Un MongoDB como capa de persistencia, al cual accede la aplicación de .NET.

El backend y la base de datos deben estar en una red llamada `lemoncode-challenge`. Para que el `front` acceda a la api, por simplicidad el backend estará expuesto en el `host` mediante el mapeo de puerto 5000.

El frontend debe estar mapeado al `host`, a través del puerto 3000.

Por otro lado, el front necesita de una variable de entorno llamada `REACT_APP_API_URL` que tendrá la URL donde está la API que consume `http://localhost:5000/api/topics`.

Del mismo modo, la API necesita de una variable de entorno llamada `MONGO_URI` donde se indica la URI de mongo (por defecto, antes de dockerizarla utiliza `mongodb://localhost:27017`). El MongoDB debe almacenar la información que va generando en un volumen, mapeado a la ruta `/data/db`.

NOTA: Para poder probar la app en local, se necesita tener instalado `.NET core` y `mongodb` NOTA: Si no quieres utilizar las variables de entorno en el proyecto de `front`, puedes dejar el código tal cual, si por el contrario quieres usarlas, cambia la línea 23 de `src/App.js`, por la siguiente:

```
let response = await fetch(`${process.env.REACT_APP_API_URL}`);
```

Pasos

General

1. Crea la red de `docker lemoncode-challenge`.

Base de datos

1. Levanta un contenedor mongo, e incluirlo dentro de la red creada anteriormente.

Backend

1. Crea una `custom image` para la aplicación de `.NET core`, como imagen base parte de `mcr.microsoft.com/dotnet/core/sdk:3.1`. Para ejecutar la `build` en `.NET core`, desde el directorio que contiene el fichero `backend.csproj`:

```
$ dotnet publish -c Release -o out
```

NOTA: Los ficheros de la build los encontraremos en la carpeta generada **out**, que hemos especificado con el flag **-o**

3. Ejecuta el contenedor anterior incluyendolo en la red **lemoncode-challenge**. Para asegurar que has tenido éxito, inspecciona la red y comprueba que el contenedor se ha añadido a la misma.

Frontend

1. Crea una **custom image** para la aplicación de **React**, como imagen base parte de **node:latest**. Para hacer que la aplicación arranque:

```
$ npm install  
$ npm start
```

2. Ejecuta el contenedor anterior incluyendolo en la red **lemoncode-challenge**, haciendo **port forwarding**, del puerto 3000.

Ejercicio 2

Ahora que ya tienes la aplicación del ejercicio 1 dockerizada, utiliza Docker Compose para lanzar todas las piezas a través de este.

- Debes plasmar todo lo necesario para que esta funcione como se espera:
 - la red que utilizan,
 - el volumen que necesita MongoDB,
 - las variables de entorno,
 - el puerto que expone la api y la web.

Además debes indicar qué comandos utilizarías para levantar el entorno, pararlo y eliminarlo.